NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IPV6 NETWORK INFRASTRUCTURE AND STABILITY INFERENCE

by

Lorenza D. Mosley

September 2014

Thesis Advisor:	Robert Beverly
Second Reader:	Garrett McGrath

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	09-26-2014	Master's Thesis 01-08-2014 to 08-30-2014

4. TITLE AND SUBTITLE

IPV6 NETWORK INFRASTRUCTURE AND STABILITY INFERENCE

5. FUNDING NUMBERS

N66001-2250-58231

6. AUTHOR(S)

Lorenza D. Mosley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of Homeland Security
245 Murray Lane SW, Washington, DC 20528

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

IPv6 deployment is increasing as IPv4 address allocations near exhaustion. Many large organizations, including the Department of Defense (DOD), have mandated the transition to IPv6. With the transition to IPv6, new techniques need to be developed to accurately measure, characterize, and map IPv6 networks. This thesis presents a method of profiling the uninterrupted system availability, or uptime, of IPv6 addressable devices. The techniques demonstrated in this study infer system restarts and the operational uptime for IPv6 network devices with a specific focus on IPv6 routers on the Internet. Approximately 50,000 IPv6 addresses were probed continuously from March to June 2014, using the Too Big Trick (TBT) to induce the remote targets to return fragmented responses. By evaluating the responses, the uptime for approximately 35% of the IPv6 addresses can be inferred.

14. SUBJECT TERMS

IPv6, Resilience, Uptime, Fragmentation, ICMP6, Security

15. NUMBER OF PAGES 59

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

IPV6 NETWORK INFRASTRUCTURE AND STABILITY INFERENCE

Lorenza D. Mosley
Chief Warrant Officer Four, United States Army
B.S., University of Maryland University College, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN CYBER SYSTEMS AND OPERATIONS

from the

NAVAL POSTGRADUATE SCHOOL
September 2014

Author: Lorenza D. Mosley

Approved by: Robert Beverly
Thesis Advisor

Garrett McGrath
Second Reader

Cynthia Irvine
Chair, Cyber Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

IPv6 deployment is increasing as IPv4 address allocations near exhaustion. Many large organizations, including the Department of Defense (DOD), have mandated the transition to IPv6. With the transition to IPv6, new techniques need to be developed to accurately measure, characterize, and map IPv6 networks. This thesis presents a method of profiling the uninterrupted system availability, or uptime, of IPv6 addressable devices. The techniques demonstrated in this study infer system restarts and the operational uptime for IPv6 network devices with a specific focus on IPv6 routers on the Internet. Approximately 50,000 IPv6 addresses were probed continuously from March to June 2014, using the Too Big Trick (TBT) to induce the remote targets to return fragmented responses. By evaluating the responses, the uptime for approximately 35% of the IPv6 addresses can be inferred.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

APNIC	Asia Pacific Network Information Center
Ark	Archipelago
AS	Autonomous System
ASN	Autonomous System Number
BGP	Border Gateway Protocol
CAIDA	Cooperative Association for Internet Data Analysis
CIO	chief information officer
CoG	Center of Gravity
CVE	Common Vulnerabilities and Exposures
DOD	Department of Defense
DoS	Denial-of-Service
EST	Eastern Standard Time
GNS3	Graphical Network Simulator
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Number Authority
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPID	Internet Protocol Identification
IOS	Internetwork Operating System
IPv4	Internet Protocol version 4

IPv6	Internet Protocol version 6
ISP	Internet Service Provider
MIDAR	Monotonic ID-based Alias Resolution
MTU	Maximum Transmission Unit
Nmap	Network Mapper
OS	Operating System
PMTU	Path Maximum Transmission Unit
PTB	Packet Too Big
RFC	Request for Comments
RIR	Regional Internet Registry
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
TSval	Timestamp Value
TCP	Transmission Control Protocol
TBT	Too Big Trick
USG	U.S. government

Acknowledgments

First and formost, I would like to thank my wife, Showanda, for her support throughout this process. Much of my time was devoted to the requirements of NPS, while, leaving many "life tasks" to her to complete. I could not have completed this journey without her.

I want to thank Dr. Robert Beverly for instructing me through both the thesis process and several demanding classes here at NPS. His desire to constantly discover new applications for existing knowledge and explore unanswered questions were an inspiration to me. His expertise, guidance, and encouragement made this work possible.

Last, I would like to thank Garret McGrath for his feedback. The ability to leverage his perspective and knowledge of Python and scripting was beneficial throughout this research.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The exhaustion of Internet Protocol version 4 (IPv4) addresses has led to an increasing number of Internet Protocol version 6 (IPv6) devices being deployed. According to Google's IPv6 adoption website, the percentage of users accessing Google via native IPv6 traffic has increased from 0.05% to 4.2% in the 4-year span from 2009 to early 2013 [1]. The number of IPv6 prefixes allocated by the Internet Assigned Number Authority (IANA) to the Regional Internet Registries (RIRs) increased from 650 in January of 2004 to 13,690 by February 2013. In the two years from 2011 to 2013, an average of approximately 200 new prefixes were allocated per month. This increase in traffic and allocations are indicators that many organizations such as universities, large companies, U.S. government (USG) agencies, and Internet Service Providers (ISPs) are transitioning from IPv4- to IPv6-based networks [2]. This transition promises to provide these entities an exponentially greater number of network addresses, increased addressing flexibility, and ease of management.

In 2011, several organizations, including Comcast, Microsoft, Google, and Facebook, contributed significantly to the understanding of IPv6 by enabling the protocol on portions of their networks. This coordinated event allowed researchers and the organizations to obtain valuable experience and measurement data from production IPv6 deployment [3]. With the adoption of IPv6 increasing, much work has been done to develop techniques to measure networks using this protocol. The Cooperative Association for Internet Data Analysis (CAIDA) Archipelago (Ark) Measurement Infrastructure, Google's IPv6 adoption, and the Asia Pacific Network Information Center (APNIC) IPv6 measurement projects, are all examples of the ongoing work focused on understanding the deployment and impact of IPv6 networks, as well as the development of techniques for measuring performance on these networks [1], [4], [5]. However, in contrast to IPv4 networks, there are relatively few tools dedicated to the measurement and analysis of IPv6 networks. Tools such as Scamper (discussed in Chapter 3), Network Mapper (Nmap), and Wireshark have implemented support for IPv6. However, IPv6 network measurement and analysis remains in its infancy.

1.1 Motivation

The large address space, flexibility, and ease of management of IPv6 is accompanied with possible security risks. Many security issues are not unique to IPv6 based networks; however, the specification for IPv6 negates some standard mitigation methods. The use of the IPv6 routing headers to direct traffic, which is similar to source routing on IPv4 networks, can be used to avoid traffic filtering devices. The IPv6 specification states that routers must be capable of processing the routing header, as opposed to IPv4 networks where source routing is disabled. Also, many IPv4 networks block Internet Control Message Protocol (ICMP) traffic to avoid many reconnaissance and enumeration attacks. However, in an IPv6 network ICMP messages are required for several important tasks (e.g., neighbor discovery). Additionally, the introduction of rogue devices could be harder to detect in large address spaces [6], [7]. Understanding the behavior of network devices in an IPv6 network is thus important.

This research investigates a specific area of IPv6 device profiling: remotely determining an IPv6 device's time since last restart and its uninterrupted availability or "uptime."

This study defines an IPv6 device as any network-attached device configured to communicate via IPv6. These devices could be routers, servers, or end-user systems. However, this research is limited to the analysis of router interfaces. An expanded evaluation that includes other device types is left to future work.

There are several intentional and unintentional events that might cause a device to restart, including power failures, system crashes, software upgrades, and malicious activity, e.g., Denial-of-Service (DoS) attacks. The ability to detect these events can aid in measuring the reliability of network devices. Reliability estimates are used to satisfy regulatory requirements, as metrics in Service Level Agreements (SLAs), and in comparing competitors. Also, detection of unscheduled or unauthorized restarts can be indicators of possible security related issues. The massive outage experienced by Time Warner Cable on August 27, 2014 [8], prompted New York Governor Andrew Cuomo to direct New York's Department of Public Service to investigate the event. Cuomo stated "dependable internet service is a vital link in our daily lives and telecommunications companies have a responsibility to deliver reliable service to their customers [9]," highlighting the importance of the need to

measure and understand reliability of this critical infrastructure.

Similar to the concept of fingerprinting, profiling uses characteristics of the device to infer information about the device. However, while fingerprinting typically seeks to identify a specific device, type of device, or Operating System (OS), profiling seeks to reveal the behavior of a device. Profiling allows measurements to not only answer what a given system may be, and how the system performs under certain conditions, but also when a device-wide event – such as a restart– occurs and ultimately help understand why the event occurred.

The goal of this study is to develop profiles for network devices to identify patterns that can be correlated to real world events. These patterns include the following: devices restarting at set intervals indicating a possible maintenance window; multiple devices restarting at approximately the same time indicating devices under the same administrative control; multiple devices in a single Autonomous System (AS), or multiple AS's restarting within a short period, indicating a possible regional or enterprise-level event. This thesis also seeks to identify anomalous behavior indicating the occurrence of one-off events. The profiles can potentially discover previously unknown correlations between routers, networks, and providers.

This research leverages the Too Big Trick (TBT) technique, developed by William Brinkmeyer *et al.* [10]. The TBT (detailed in Chapter 3) was developed to work in an IPv6 environment. Expanding the techniques developed in this study to measure uptime in IPv4 environments is the subject of future work.

1.2 Department of Defense Applicability

As the Department of Defense (DOD), other governments, and nations transition to IPv6, the techniques developed in this study can be used to collect data about critical infrastructure and devices of interest. The information collected can be used for defensive or offensive operations. In both types of operations these techniques can aid in Center of Gravity (CoG) analysis, helping to identify critical nodes by measuring the effects on the networks if and when a device is restarted. Additionally, these techniques may reveal the current security level of a device. If a security update was released and the device shows no indication of being restarted, the inference would be that the update was not applied. Of-

fensive cyber operations could use this information for targeting purposes, while computer network defense activities could use the information to verify that security patches are being applied. Also, both entities could use these techniques as additional tools to determine if an attack on a router or other network device was successful.

1.3 Research Questions

The questions this study aims to answer are:

1. Can IPv6 fragment identifiers reveal reboots and system uptimes?
2. Do cycles or anomalies exist? Cycles refer to events that occur at regular intervals, for example a device that reboots on the 15th of every month. While an example of an anomaly is a device that has not previously restarted restarting multiple times in a single day.
3. Are there correlations between seemingly unrelated systems?

1.4 Summary of Major Contributions and Findings

The major contributions and finding of this research are as follows:

- Validation of the use of induced fragmentation via the TBT to identify router restarts.
- Introduction of a behavior-based profiling method of network measurements.
- Internet-wide probing of more than 49,000 candidate IPv6 addresses of which approximately 30% display identifiable restart behavior.
- Identification of approximately 5% of restarted addresses that are active for short periods and present a large number restarts.

1.5 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2 provides an overview of the fundamentals of IPv6 and a review of prior research that was used as the basis for this thesis.
- Chapter 3 outlines the methodology used in the thesis to include virtual lab configuration, data set validation, and live network testing.

- Chapter 4 details the results derived from the analysis of data captured during this study.
- Chapter 5 discusses conclusions based on this research and provides recommendations for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Related Work

The profiling approach, described in Chapter 1, can enhance the effectiveness of measurements by eliminating variables such as interface or path failures. This benefit is achieved by eliciting information from the device's control plane. Changes in the control plane relate to the device as a whole independent of the state of any single interface. Implementation differences between IPv4 and IPv6 have enabled new measurement techniques while rendering some existing approaches obsolete. Techniques that involved probing the entire range of possible addresses, such as the one used by Durumeric *et al.* in their study of the Hypertext Transfer Protocol Secure (HTTPS) ecosystem [11], would be impractical in an IPv6 network.

This chapter covers a brief overview of relevant IPv6 (§2.1.1) characteristics. Also, existing measurement techniques that form the groundwork for this study are explored.

2.1 Background

IPv6 deployment continues to gain momentum as IPv4 address allocation nears exhaustion. The chief information officer (CIO) for the federal government released a memorandum in September 2010 informing all USG agencies to transition to IPv6 [12]. The memorandum included milestones for both public facing services and internal client applications that communicate with the Internet or support enterprise networking to employ native IPv6 communications [12]. IANA allocated the last five available full Class A network blocks to RIRs in February 2011 [13]. Starting in 2012, customers of Comcast's home Internet services were issued IPv6 addresses [14]. In April 2014, IANA announced that a single Class A address block remained to service any future IPv4 request on a "First in, First out" basis [15].

As the number of devices that connect to the Internet continues to increase, the adoption of IPv6 is becoming a necessity. Additionally, the need for accurate measurement and analysis of IPv6 networks becomes more important as more devices use IPv6 addresses natively. Network measurement and analysis of IPv6 networks remains in its infancy, mainly

because many widely used techniques developed for IPv4 networks do not work in IPv6 environments.

2.1.1 IPv6 Fundamentals

IPv6, the successor of IPv4, exponentially increases the number of possible IP addresses from 2^{32}, approximately 4.2 billion, to 2^{128}, representing more than 340 undecillion (*340 followed by 36 zeros*) addresses. To make IPv6 addresses easier to represent, the dotted decimal IPv4 format (111.222.333.444) was replaced with a new hexadecimal colon delimited format (AAAA:BBBB:CCCC:DDDD:EEEE:FFFF:0000:1111).

The IPv6 header (Figure 2.1) is simpler than the of IPv4 (Figure 2.2). The removal of many of the fields in the IPv4 header allows IPv6 to have a fixed header length of 40 bytes verses the variable 20 to 60 bytes of its predecessor. Some of the information represented by the deleted fields is included in IPv6 extension headers, while some information no longer has any relevance. Of particular interest to this work, are the identification, flags, and fragment offset fields. In IPv6, these fields are incorporated into an extension header known as the fragment header.

2.1.2 IPv6 Fragmentation

Packets are fragmented when their size exceeds the Path Maximum Transmission Unit (PMTU) of any link through which the packet must travel. Since the PMTU can change as packets are routed along different paths to their destination, most nodes use PMTU discovery to ensure the packet size does not exceed the PMTU of any route to its destination. If PMTU discovery is not used, nodes set the size of the packets sent to the minimum link Maximum Transmission Unit (MTU) value allowed in the IPv6 specifications, 1280 bytes [6], [17]. When using the PMTU discovery process a node estimates the MTU of a given path. If the actual MTU for any segment of the path is less than the estimate, an ICMP Packet Too Big (PTB) message is sent back to the node by the router containing the MTU for the link and as much of the original packet as possible without exceeding 1280 bytes. Once received, the MTU in the packet too big message is cached by the node, for no less than five minutes. All packets larger than the PMTU generated by the node using that path are fragmented to a size that does not exceed the cached value [17], [18]. The

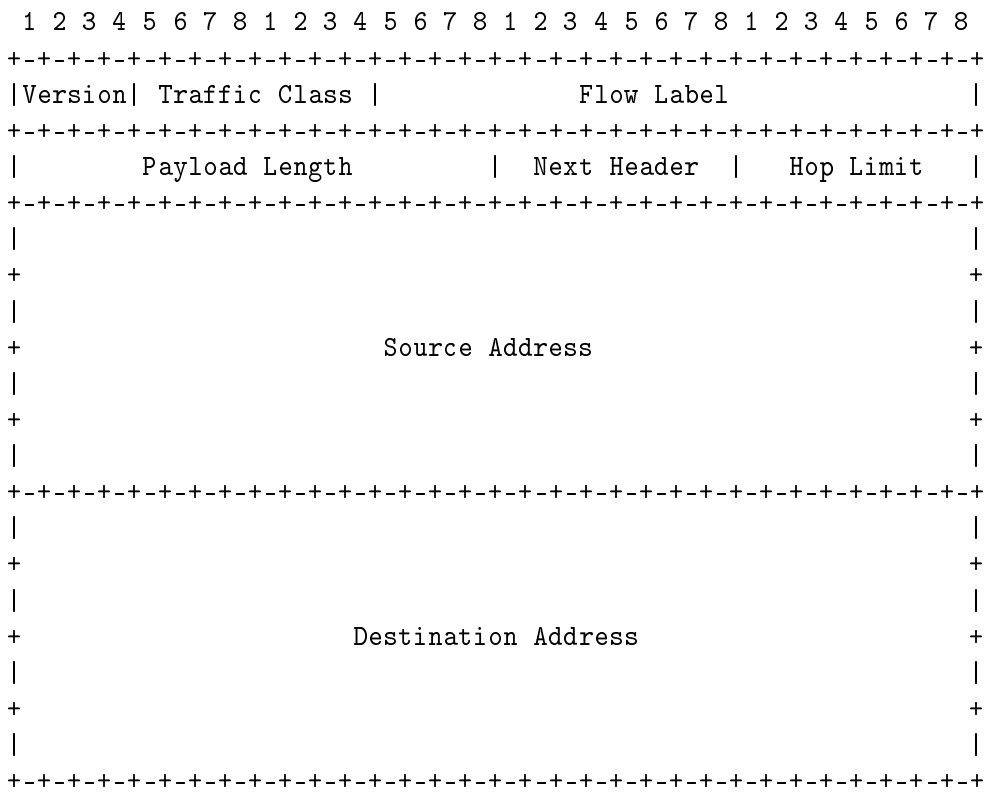

Figure 2.1: IPv6 Header, from [6].

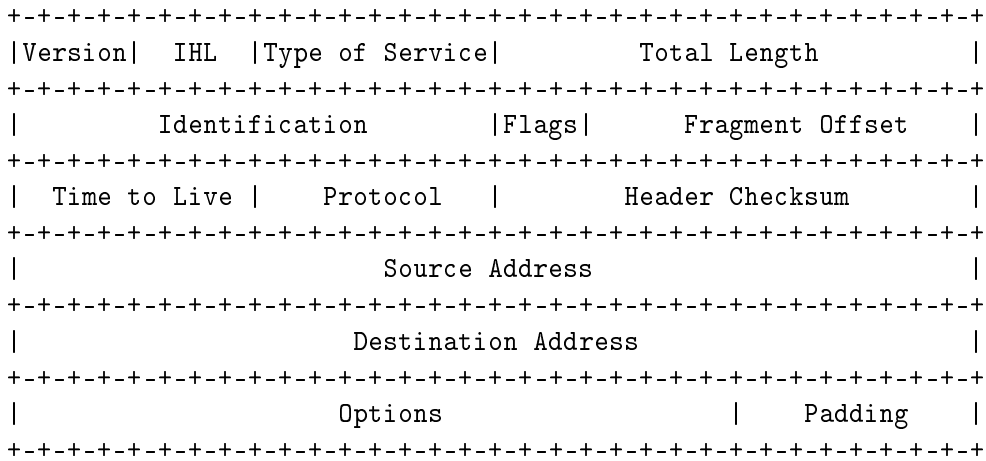

Figure 2.2: IPv4 Header, from [16].

fragmentation extension header is added to all fragments of the original packet.

In IPv6, fragmentation is not done by the routers. Instead, fragmentation is the responsibility of the sending host. Removing in-transit fragmentation eliminates the processing overhead needed to deal with fragments from the routers and helps make forwarding more efficient. When a packet needs to be fragmented, the sending host will add a fragment header to the packet. The presence of this header is indicated by the value 44 in the next header field of the header immediately preceding the fragment header [6].

The fragment header (Figure 2.3) is composed of six fields. A detailed description of each field from Request for Comments (RFC) 2460 is provided below [6]:

- *Next Header* - Uses the same values as the protocol field in the IPv4 header. Identifies the initial header type in the fragmentable part of the original packet. The fragmentable part of a packet refers to the packet minus the IPv6 header and any extension headers that needs to be processed by any nodes en route to the destination.

- *Reserved* - A reserved 8-bit field. At transmission the field is initialized to zero, and is ignored by recipients.

- *Fragment Offset* - An unsigned integer representing the offset of data relative to the beginning of the fragmentable part of the original packet.

- *Res* - A reserved 2-bit field. At transmission the field is initialized to zero, and is ignored by recipients.

- *M flag* - This field is 1-bit long; when set to 1 the field indicates more fragments follow, and last fragment when set to 0.

- *Identification* - A 32-bit field that contains a value generated by the source of the packet to identify all fragments that comprise the original packet. This value must be different than any other recently generated for packets with the same source and destination address.

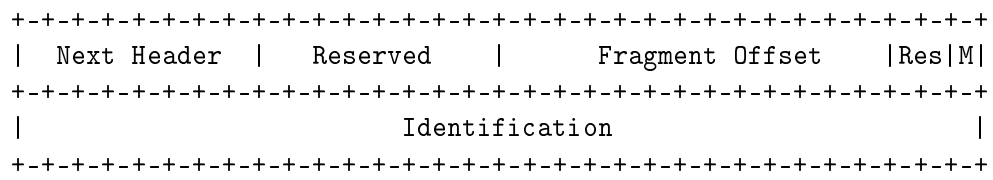

Figure 2.3: IPv6 Fragment Header, from [6].

The 32-bit size of the identification field in the fragment header is of particular importance to this thesis. Approximately 4.2 billion values can be represented in this field as opposed to only 64k in the 16-bit identification field of the IPv6 header [16]. Additionally, the PMTU process minimizes fragmentation in an IPv6 network, effectively ensuring the fragment identification field is seldom used. These two factors eliminate the problems of identification number overflow and wrapping that would otherwise make this field unsuitable for use in this study.

The PMTU discovery process, and the establishment of a 1280-byte minimum MTU value, is meant to reduce, but not eliminate, fragmentation in IPv6 networks. Fragmentation is a normal response to changes in network conditions; however, deliberate fragmentation is used for various reasons. Deliberate fragmentation is not new in IPv6 networks. Similar to IPv4, deliberate fragmentation in IPv6 networks can be used for Intrusion Detection System (IDS) evasion [19] and OS fingerprinting [20]. Deliberate fragmentation is also used for alias resolution (§2.2) to identify a single device with multiple IPv6 addresses assigned.

2.2 Related Work

Most prior stability measurement work is concerned with end-to-end path measurements [21]–[23]. Current stability measurement techniques do not focus on the endpoints. These techniques rely on variants of tools, such as traceroute and ping, to infer congestion or jitter along the path between the endpoints. Much of the work targeted at endpoints is in the area of alias resolution. Alias resolution determines if multiple layer 3 addresses belong to a given router. Many of the techniques pioneered to perform alias resolution measurements are used as the basis of this work.

2.2.1 Alias Resolution techniques

Many tools and techniques have been created to perform alias resolution including Ally [24], Monotonic ID-based Alias Resolution (MIDAR) [25], RadarGun [26], and speedtrap [27]. Fundamentally, these techniques rely on the fact that the multiple interfaces of a router use a common counter when generating fragment identifiers. Speedtrap is the only technique in this list that is used on IPv6 networks. RadarGun and speedtrap provide a foundation for this thesis work.

RadarGun introduced the concept of measuring the rate a router's fragmentation counter increases to determine the natural "velocity" of the counter. Through observation, a researcher can infer that interfaces with the same velocity are in fact aliases. This helped overcome the O(n^2) problem that limited Ally [26].

Speedtrap expanded on the authors' earlier work where the TBT was introduced [10]. The TBT is a technique used to cause an IPv6 enabled device to send fragmented packets in response to a crafted PTB message (illustrated in Chapter 3). Speedtrap also demonstrated that IPv6 routers, in contrast to IPv4 routers, do not have a natural velocity. The lack of a natural velocity allows for more accurate measurements using the Internet Protocol Identification (IPID) generated by IPv6 devices. Additionally, Speedtrap identified several different ways that routers respond to PTB messages: with monotonic incremental IPIDs, random values, or by sending no fragments or becoming unresponsive [27].

2.2.2 System Restart and Operational Time Inference

Many operational time (uptime) measurements require local access and are not focused on individual network devices, but rather the path between the devices. Network management tools, such as the Simple Network Management Protocol (SNMP), can be used to remotely determine the uptime of devices; however, these tools require some level of administrative access to the devices. This study assumes no administrative access to devices. The Nmap tool (discussed next) includes a technique to remotely infer the operational times of end-user devices.

The measurement of operational times for home broadband networks is gaining greater importance, as more homes become more reliant on connecting to the Internet for services such as banking, emergency assistance, and home phone. The measurements for these

networks rely on collecting data locally from end-points within the homes. The focus of these measurements are service interruptions and packet loss from the perspective of the local device, a cable modem for example. [28]

Router-level restarts are sometimes addressed in studies focused on the stability of Border Gateway Protocol (BGP). These studies illustrate how a routing update caused by a link failure can trigger multiple updates in BGP; however, router restarts are one of many possible causes for link failures [29]. Studies of this type do not specifically measure individual router behavior, but instead focus on effects routers have on the overall BGP infrastructure.

Nmap, an open source network discovery and security auditing tool, attempts to remotely infer how long a end-user system has been operational as part of the tool's OS discovery process. Nmap uses the Transmission Control Protocol (TCP) timestamp options to estimate the number of seconds from the last system reboot. The TCP timestamp option is a 10-byte field that can be specified as part to the TCP header [30]. The system's current clock value, stored in the Timestamp Value (TSval) subfield, a 4-byte field within the timestamp option field, is used as the basis of system operational time estimations. The relatively small size of the TSval affects the accuracy of this calculation due to value wrap around. Accuracy is also lessened because the TSval field is not initialized to a standard value, such as zero when reset. [31]

The Netcraft service uses a similar technique to determine the uptime of websites [32].

This thesis attempts to bridge the gap in stability measurements by introducing techniques to remotely infer restarts and operational time at the individual router level.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

This study involves analyzing IPv6 responses to specific stimuli to determine what information about target systems can be inferred. Crafted traffic was transmitted from a probing system to target systems to elicit a specific response containing IPv6 fragment identification numbers. Careful analysis of the fragment ID numbers can reveal a change in the state of many devices, specifically whether the device was rebooted since it was last probed.

The manipulation technique used for this thesis is the induced fragmentation process outlined in the paper "IPv6 Alias Resolution via Induced Fragmentation" [10] and termed "TBT." This technique involves sending a 1300-byte ICMP6 echo request (ping) to a target to ensure that the target is alive and responds to ICMP6 echo requests. If the corresponding echo response is received, the prober sends a PTB message with a small MTU (e.g., 1280 bytes) to the target. The target (or target's interface) caches the MTU size specified in the PTB message as the PMTU to the prober. The TBT technique causes the target to fragment any packet destined for the source of the PTB message that is larger than the size specified, as shown in figure 3.1.

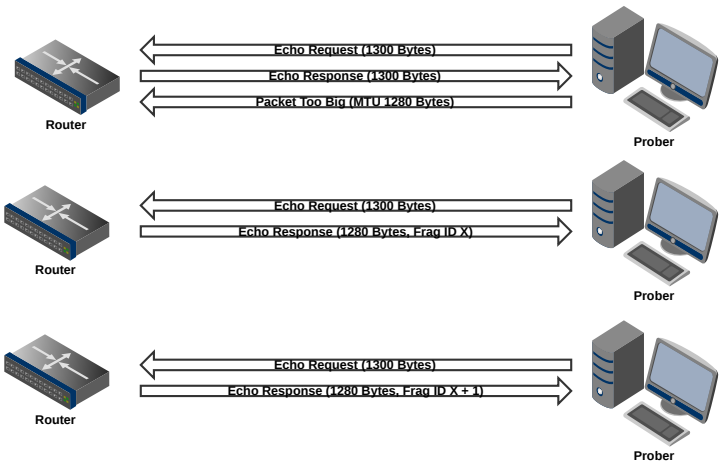

Figure 3.1: Too Big Trick (TBT)

It is important to note that TBT exercises a router's control plane functionality, i.e., the IPv6 stack of the router itself, rather than any forwarding functionality. TBT is particularly effective because network devices (routers and servers) in an IPv6 architecture are normally not the originators of fragmented traffic, the IPv6 specification states that the originating entity is responsible for fragmentation [6].

This chapter describes the phased methodology used in this thesis.

3.1 Laboratory Testing

Before performing live Internet measurements, this study first used a virtual lab consisting of a probing system, network routers, and end-host systems to assist in understanding how IPv6 nodes respond to the probing technique used in this thesis. A combination of VirtualBox and Graphical Network Simulator (GNS3) [33] was used to build the virtual environment shown in Figure 3.2. The virtual guest systems consisted of two routers, running different versions of the Cisco Internetwork Operating System (IOS); end-hosts running Ubuntu Linux, Windows 7, CentOS Linux, and OpenBSD end-hosts running Ubuntu, Windows 7, CentOS, and OpenBSD; and a Debian Linux probing system. Scamper, a multi-purpose packet-prober [34], was the primary tool used for sending and collecting probe data. The probing system used Scamper and Linux shell scripts to send probes to the destination host to emulate the TBT. Testing in this environment allowed the probing and data collection process to be refined and verified. First, laboratory testing was used to verify the Scamper tool was able to implement the TBT. Also, through testing and observation, the number of probes to send to each host per round was determined. Four probes per host was revealed to be ideal to collect enough information to analyze the host while minimizing the time per round.

In addition, the behavioral characteristics of the various guest OS's outlined by Brinkmeyer were verified [35]. The Windows and Linux systems returned monotonically increasing sequential fragment identifiers in response to probes. Both OS's also reset the counters to a base value after a restart. The only difference is that the Windows system reset the counter to zero while the Linux systems reset the counter to an predetermined arbitrary value. The OpenBSD system returned random values in response to probing and restarts. These results are consistent with the finding in the Brinkmeyer work. Both IOS systems

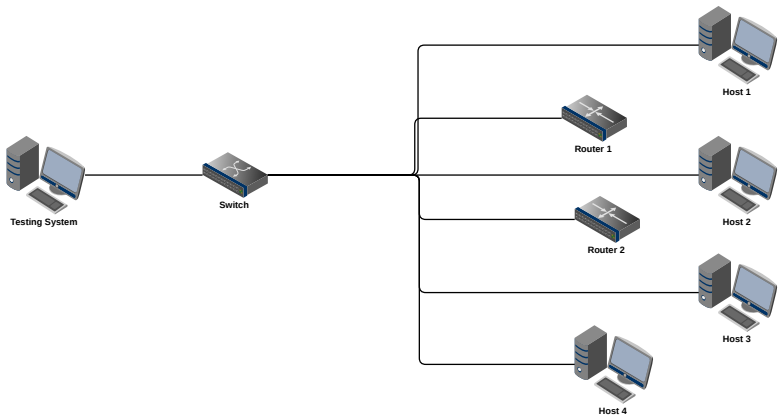

Figure 3.2: Testing Lab Layout

returned sequentially increasing fragment identifiers in response to probes, and reset the initial identifier value to 1 after restarts.

3.2 Data Set Validation

The CAIDA IPv6 Topology Datasets from January and February 2014 were used as the source of possible candidate router addresses for this study. These Datasets were created using Scamper to preform continuous traceroutes from CAIDA's globally distributed Ark monitors [36]. The 968 trace files collected by CAIDA during this two-month period were parsed in order to compile a list of candidate router interface addresses. First, all the records were combined and normalized to remove any duplicate addresses. Next, any IPv6 link-local addresses (fe80::/10) were filtered out. These addresses are equivalent to 169.254.0.0/16 addresses in IPv4. The IPv6 link-local address is used for automatic addressing, neighbor discovery, or in the absence of a router for single link connections. Packets sourced from this address range should not be forwarded by routers [37]. The parsing process produced a list of 52,154 candidate addresses. Whois searches, using the Team Cymru community service [38], revealed these addresses spanned 3,100 AS's, registered in 125 countries.

3.2.1 Candidate testing

The next step in the data set validation process was testing the suitability of each candidate address. An address was considered suitable for this thesis if the address consistently re-

sponded to ICMP6 echo requests generated by Scamper. Suitability testing was conducted from an Internet attached Linux system configured with the Scamper tool and native IPv6 networking. The first phase of testing used the system's built-in ping utility. All systems in the candidate list were sent echo requests at random intervals over a three-day period; each address was sent 10 echo requests per interval. Systems that failed to respond to at least one request, indicating the system was completely unreachable, were removed from the list. Next, the Scamper ping command was used to probe the modified list of addresses at random intervals for an additional three days. Systems failing to respond to at least one request during this phase of testing were removed from the list. Approximately 95.5% of all candidates tested were suitable, producing a final list consisting of 49,823 responsive node addresses.

3.3 Internet-Wide Testing

A single Internet vantage point was used to probe all addresses and collect trace data every six hours from March 5 to July 31, 2014. A random permutation of the address list was used for each round of probing. During each round 40 addresses were probed simultaneously. A probing round consisted of preforming the TBT to set the PMTU to 1280 bytes, followed by three 1300 byte echo requests (refer to fig.3.1). Scamper was used to generate the traffic in each round. The traffic was saved in Scamper's native file format as timestamped trace files [34]. Approximately 1.2 million probes were sent over the period covered by this study. The traffic from these probes produced 5.8 gigabytes of data, saved in 525 trace files.

The trace files were parsed to collect the fragment identifiers for each address. Addresses were separated into three general groups based on the fragment identifiers; incremental, random, or not fragmented. Additionally, a special group of addresses that consistently stopped responding after receiving the PTB message was identified. Addresses in the incremental group returned fragment identification numbers that increased in a predictable sequential pattern, e.g., if identifier x was received the next identifier received was $x+1$. Addresses in the random group returned fragment identification numbers with no discernible sequential pattern, i.e., given identifier x the next identifier could not be predicted with any degree of confidence. The no fragments group responded to probes but did not return fragmented packets. This behavior was attributed to the possible existence of a device along

the path between the probing system and target address that reassembled the fragmented responses, or a device along the path blocking the PTB message. Lastly, the special group consisted of addresses that responded to the initial probes but went silent after the PTB message was sent. Table 3.1 shows the distribution of addresses relative to each group.

Table 3.1: Addresses Grouped by Identifier

Group	Addresses	Percentage
Incremental	17,227	34.6%
Random	16,265	32.6%
No Fragments	12,224	24.5%
No Response post PTB	4,107	8.3%

The categorization of each address was performed using Scamper's sc_speedtrap, a parsing tool designed to analyze fragmentation identification numbers based on the *Speedtrap* algorithm (see §2.2). Also, manual evaluation of the responses from a random sampling of approximately 200 addresses per month was performed to verify the results produced by sc_speedtrap. The identifiers produced by addresses in the random group did not reveal any information that could be used to infer system restarts. The behavior of addresses included in the incremental group proved more suited for researching the viability of the techniques used in this study.

This study evaluated each address as an individual system. Multiple addresses may belong to the same physical device; however, alias resolution is not an objective of this research.

3.3.1 Probing Validation

The probing system was checked throughout the duration of this study to ensure that probes were sent during each of the defined rounds. System availability for the probing systems was tested using system utilities, such as the uptime command, at different times throughout the testing period. The ping6 command was used to test system connectivity, verifying that the probing system could send and receive IPv6 traffic. The completeness of the probing was also verified by periodically parsing a sample of the trace files to ensure the file contained entries corresponding to each address in the list; once verified, these files were

categorized as "known goods." Additionally, the file size of each trace file was compared to the size of a known good file as a secondary check. Each trace file should contain the same number of entries, in the same format, therefore, there should be a variance of no more than 1% in the file size. Files that failed these checks were evaluated for usability and discarded when necessary.

3.4 Reboot Algorithm and Operation Time Calculation

The determination of system restart was accomplished by comparing fragment identifiers collected over a period of time. Algorithm 1 provides the system reboot detection pseudocode. To determine if a system restarts, the TBT is preformed to induce fragmented responses from the target system. Next, a series of probes are sent to the target. The responses from each probe are parsed to extract the fragment identifiers and timestamps (timestamps are recorded in Epoch time). The most recent identifier (ID_{curr}) and timestamp (TS_{curr}) are compared to the preceding identifier (ID_{prev}) and timestamp (TS_{prev}). If the current timestamp is more recent than the previous timestamp, and the current identifier is less than the previous identifier, a restart is indicated.

Algorithm 1 *IPv6 Reboots*: Determine whether an IPv6 address restarts

```
    ID_prev ← 0
2:  TS_prev ← 0
    send(TBT)
4:  send(echo)
    for i in range(4) do
6:      ID_curr ← echo(ID[i])
        TS_curr ← echo(TS[i])
8:      if TS_curr > TS_prev and ID_curr > ID_prev then
            TS_prev ← TS_curr
10:         ID_prev ← ID_curr
            Continue
12:     end if
        if TS_curr > TS_prev and ID_curr < ID_prev then
14:         return True
            TS_prev ← TS_curr
16:         ID_prev ← ID_curr
        end if
18: end for
```

Once a system restart was identified, the formula (TS_{prev} - TS_{curr})/86,400 was used to de-

termine the system's operational time prior to or between restarts. The formula converts the Epoch time used for timestamps to a number of days. In the formula, the previous timestamp represents the starting time (e.g., beginning of time period, or last recorded restart), the current timestamp represents the ending time, and 86,400 represents the number a seconds in a day.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Analysis

This chapter presents detailed results from the analysis of traffic traces captured in this study. The analysis focuses on the 17,227 addresses from the incremental group identified in Chapter 3, Table 3.1. The addresses were evaluated based on two factors: fragment identifier and time. 902 addresses ($\approx 5\%$) displayed evidence of a 25% to 300% higher number of restarts, compared to other addresses. These addresses are evaluated separately in Section 4.2, because this behavior being the result of actual restarts was unlikely.

Figure 4.1 summarizes the restart behavior of the remaining 16,325 addresses. Approximately, 64% of addresses did not restart during this study. The following sections concentrate on analyzing the 5,933 addresses that indicated restarts.

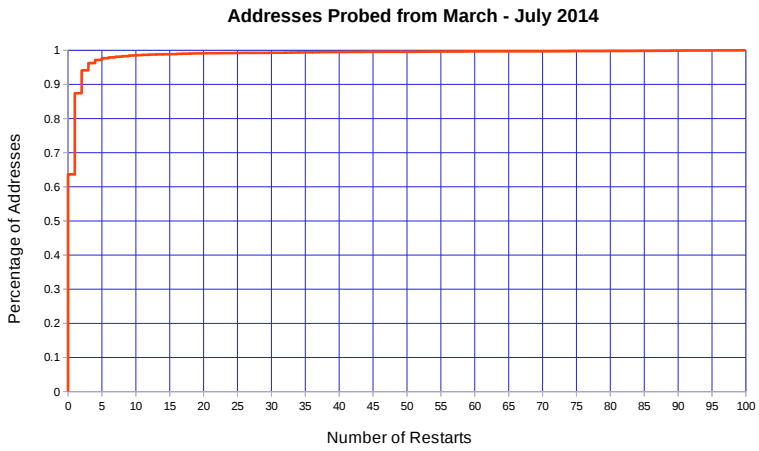

Figure 4.1: Cumulative distribution of restarts detected among the 16,325 addresses evaluated

4.1 Analysis of Restarted Addresses

The fragment identifiers of the addresses analyzed in this section increase monotonically; a decrease in the fragment identifiers indicate a restart. Laboratory testing, Section 3.1, demonstrated that when a device restarts, the fragment identifier is reset to an initial value, e.g., 0 or 1. Addresses were evaluated against this property to determine the number of times the addresses restarted. The evaluation revealed approximately 66% of the addresses that experienced a restart restarted once, while 85% restarted once or twice, and 95% of addresses restarted fewer than 6 times during the time period March to July 2014 (see Figure 4.2).

The number of restarts presented are considered the lower bound of possible restarts. This research collected data 4 times daily in 6 hour windows, therefore multiple restarts in the same window for a given address would be missed.

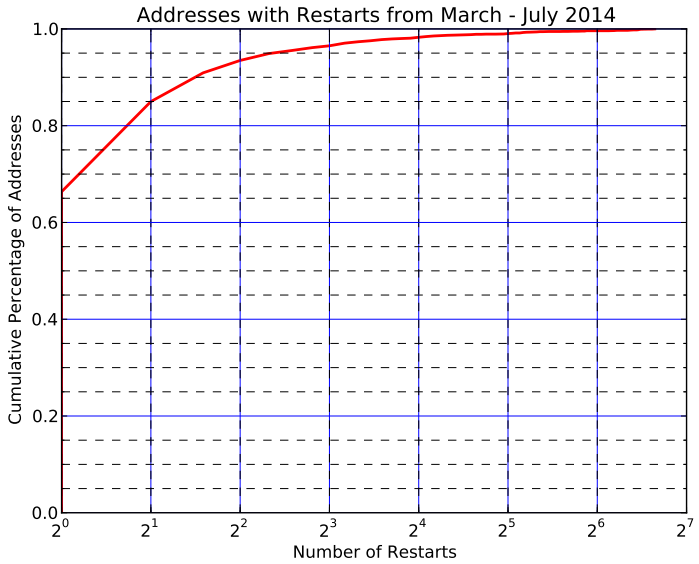

Figure 4.2: Cumulative distribution of addresses that restarted at least once from March to July 2014

4.1.1 ASN Mapping

Whois searches using the Team Cymru community service [38] were used to map each address to an Autonomous System Number (ASN). The addresses mapped to 1,315 AS's, spanning 103 countries. The country information was gathered from the whois records returned. The MaxMind GeoIP2 database [39] was used to verify the location of approximately 90% of restarted addresses, no information about the remaining addresses was found in the database. The number of addresses in a given AS that experienced one or more restarts ranged from 1 to 427. Figure 4.3 displays the distribution of addresses among AS's. The distribution shows 95% of AS's contain less than 10 restarted addresses.

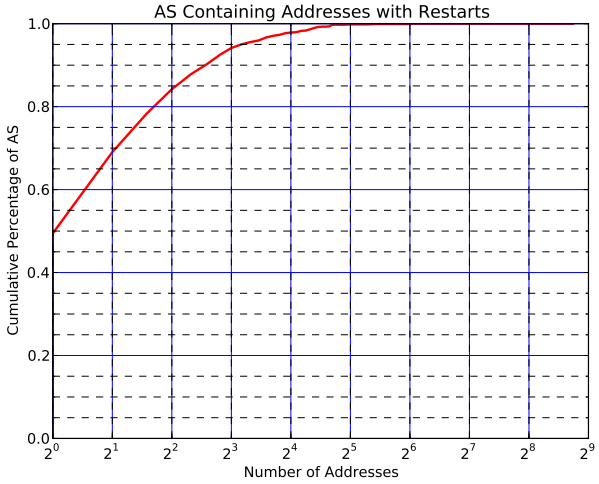

Figure 4.3: Cumulative distribution of addresses that restarted per AS from March to July 2014

The AS's were ranked based on two factors: the number of restarted addresses associated with the AS, and a normalized fraction of the total addresses from the original 16,325 for each AS. The normalized fraction removes the bias caused by the disproportionate distribution of addresses among AS's in our dataset.

Table 4.1 enumerates the top 20 AS's representing 30% of restarted addresses. Hurricane Electric contained the most addresses experiencing a restart, however, due to the large total number of Hurricane Electric addresses in our dataset, it is ranked sixth relative to other AS's based on the normalization fraction. This evaluation was restricted to AS's containing 25 or more addresses.

Table 4.1: Top 20 AS Containing Restarted Addresses.

AS Number	Organization Name	Number of Addresses	Percentage	Normalized
55430	Starhub Internet Pte Ltd	93	1.57%	97.89%
19271	Peak 10	74	1.25%	97.37%
12989	Eweka Internet Services B.V.	46	0.78%	80.70%
30071	TowardEX Technologies International, Inc.	53	0.89%	73.61%
22822	Limelight Networks, Inc.	58	0.98%	52.73%
6939	Hurricane Electric, Inc.	427	7.20%	49.36%
4826	Vocus Connect International Backbone	39	0.66%	48.75%
5580	TripartZ B.V.	68	1.15%	40.96%
1200	Amsterdam Internet Exchange B.V.	71	1.20%	40.34%
174	Cogent Communications	150	2.53%	39.47%
7922	Comcast Cable Communications, Inc.	130	2.19%	38.81%
1299	TeliaSonera International Carrier	66	1.11%	38.15%
6695	DE-CIX Management GmbH	54	0.91%	36.73%
3257	Tinet SpA	63	1.06%	34.05%
3549	Level 3 Communications, Inc.	66	1.11%	33.17%
5459	London Internet Exchange Ltd.	44	0.74%	32.12%
2914	NTT America, Inc.	51	0.86%	20.82%
3356	Level 3 Communications, Inc.	122	2.06%	20.33%
11427	Time Warner Cable Internet LLC	44	0.74%	19.47%
2516	KDDI CORPORATION	55	0.93%	18.27%

Further, analysis revealed that though the addresses were registered in 103 counties, 85% of restarted addresses map to the 20 countries shown in Table 4.2. Approximately five times more addresses map to the United States than any other country; however, Ukraine, Singapore, and Russia contain more restarted systems per capita.

Table 4.2: Top 20 Countries with Restarted Addresses

Country	Addresses	Percentage	Normalized	Country	Addresses	Percentage	Normalized
Ukraine	96	1.62%	55.17%	Czech Republic	85	1.43%	34.69%
Singapore	139	2.34%	48.77%	Austria	102	1.72%	34.58%
Russia	191	3.22%	44.11%	France	138	2.33%	34.50%
Italy	91	1.53%	39.39%	United States	2245	37.84%	34.34%
Romania	60	1.01%	39.22%	India	57	0.96%	34.34%
Netherlands	316	5.33%	38.68%	Germany	445	7.50%	34.15%
Brazil	74	1.25%	38.54%	Australia	173	2.92%	33.99%
Indonesia	66	1.11%	36.67%	Great Britain	223	3.76%	29.46%
Sweden	184	3.10%	36.08%	Switzerland	88	1.48%	22.39%
Canada	119	2.01%	35.52%	Japan	169	2.85%	19.07%

4.1.2 Time-Based Analysis

Throughout this study each restart detected for an address was tagged with a Epoch timestamp indicating the time of restart. Each timestamp was used to calculate the time between restarts for each address. These calculations are limited to the six hour span between probing windows. Also, multiple restarts between probing would introduce errors in this calculation; if an address experienced *n* restarts only the *nth* restart would be detected, creating a possible error rate of 1 - $\frac{1}{n}$.

Table 4.3 gives an overview of general time ranges. The table only includes addresses that restarted more than once during this study. The time ranges in the table represent the average time between restarts for a given address. Addresses with a single restart do not provide sufficient data points to determine the average time between restarts. The range 1 - 30 days contains double the number of addresses of all other ranges combined. This observation may suggest conformity to the commonly used 30 day maintenance cycle.

Table 4.3: Average Days Between Restarts.

Day Range	Addresses	Percentage
1 - 30 Days	1465	77.9%
31 - 60 Days	376	20.0%
> 60 Days	40	2.1%

Table 4.4 summarizes the number of months in which a particular address restarted. The number of months indicate the total number of unique months from the set of measured months in which the address experienced a restart. The majority of addresses, 72.37%, experienced restarts in only 1 of the 5 months during this study. The evaluation revealed that fewer than 2% of addresses restarted at least once in every month during this study. Overall, only 9.27% of addresses experienced restarts in 3 or more of the months of this study.

Table 4.4: Months Restart Detected per Address

Number of Months	Addresses	Percentage
1	4294	72.37%
2	1089	18.35%
3	311	5.24%
4	123	2.07%
5	116	1.96%

Each address was evaluated to determine the number of days between the address' last restart and the end of the study (July 31, 2014). Figure 4.4 Illustrates the distribution of operational times for each address. Approximately, 29% of addresses restarted within 30 days prior the end of the study, with a slight plateau between 27 and 30 days. An additional 20% of addresses restarted within 60 days prior the end of the study. More than 50% of addresses restarted between 62 and 150 days prior to the end of the study, with an $\approx 5\%$ spike at 128 days.

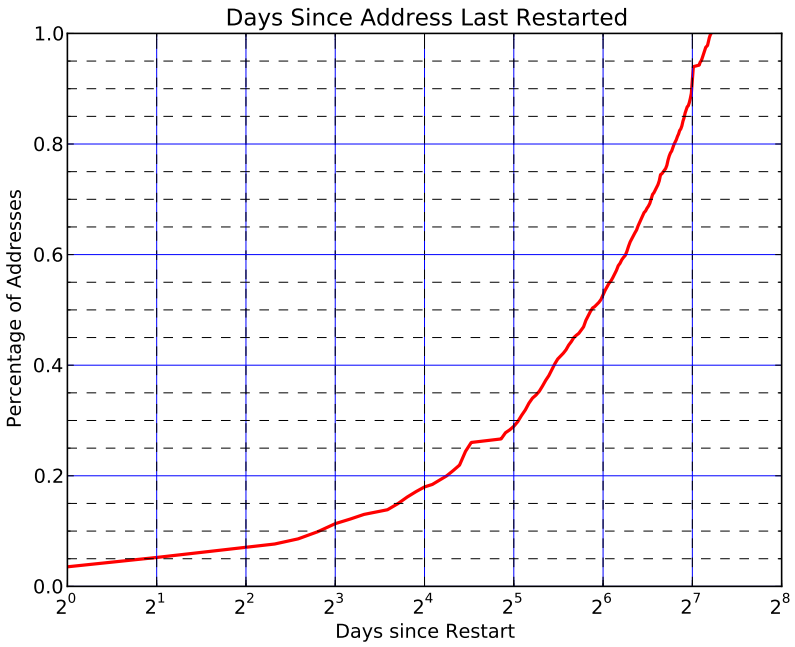

Figure 4.4: Days from Last Restart to End of Study

This study used four probing windows to collect data. The probing windows were as follows (Eastern Standard Time (EST)):

- Window 1: 00:00 - 05:59
- Window 2: 06:00 - 11:59
- Window 3: 12:00 - 17:59
- Window 4: 18:00 - 23:59

The fewest restarts occurred during probing windows two and three, 23% and 21% respectively. Approximately 29% of restarts happened during window 4; while, 26% of restarts occurred in window 1 (see Table 4.5).

Table 4.5: Restarts Per Probing Windows

Probing Window	Host	Percentage
1	1566	26.4%
2	1371	23.1%
3	1269	21.4%
4	1727	29.1%

4.2 Anomalous Addresses

There were 903 addresses, spanning 416 AS's, registered in 55 countries, identified during this study that appeared to restart an unusually large number of times. Further evaluation revealed the addresses appeared to restart between *every* probing round. Approximately 99% of these addresses returned at least two alternating sets of fragment identifiers (x, x+1, x+2, y, y+1, y+2, x, x+1, x+2, ...). Only 8 addresses, belonging to 6 AS's, consistently responded with the same sequence of fragment identifiers. This behavior indicates that different devices responded to probes sent to a single address. The number of alternate patterns varied from 2 to 5 across different addresses. Additionally, the frequency of alternations was inconsistent across different addresses, or when observing a single address. The addresses were probed in 2 hours, 1 hour, 30 minute, and 15 minute intervals in an

attempt to better characterize the cyclical identifier behavior and its dependence on the probing.

The additional rounds of probing were performed in 24 hour cycles. During each cycle a different set of addresses responded to probing, suggesting some addresses were active for only short periods of time. The responsive devices continued to respond with repetitive fragment identifiers.

This behavior seemed unrelated to addresses experiencing restarts, but instead indicated the presences of load balancing equipment, or perhaps security devices that maintain state based on flows.

4.3 Observations

This study focused on determining restart behavior for addresses with incremental fragmentation identification numbers. However, the initial evaluation process revealed additional information worth mentioning.

- **Device Replacement** - Two addresses returning random identifiers at the beginning of the study, began continuously returning incremental identifiers later in the study. This change in behavior suggests the device associated with these addresses was replaced with a different device.
- **Spike in Restarts** - On March 24th, double the number of addresses restarted compared to any other day during this study. This activity may correlate with a vulnerability announcement related to the software present on many network devices. The MITRE corporation published a Common Vulnerabilities and Exposures (CVE) alert detailing a DoS vulnerability in network devices approximately 30 days earlier. The manufacturer of the devices publicly released detailed information about the vulnerability on March 26th. Another, significant spike occurred on July 8. This spike also follows the publication of a CVE alert approximately 30 days earlier.

  Figure 4.5 illustrates the number of restarts seen per day during this study. Restart activity was highest during the first 30 days of the study. The majority of restart activity remained at or below the median (121) level of restarts, until the final 30 days of probing.

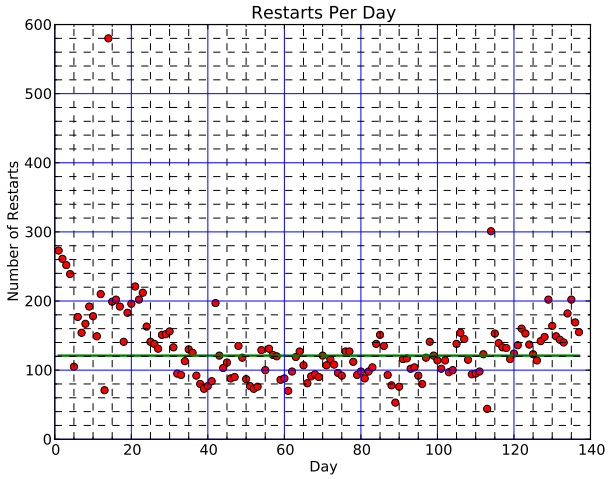

Figure 4.5: Number of restarts per day during this study. The green line represents the median number of restarts per day, 121.

This analysis revealed that overall most devices associated with the IPv6 addresses tested were stable, based on patterns of behavior. The predicable behavior of addresses was determined over time, making identifications of anomalies possible.

Also, the data discovered during this analysis allowed profiles for individual as well as groups of devices to be constructed. For example, Table 4.6 presents a basic profile of the addresses in AS 55430 built from data collected in this study (the actual IPv6 addresses were replaced by letter in the table to anonymize the data). Additionally, the table illustrates addresses A and B, and addresses D and E demonstrate similar overall behavior, indicating a relationship beyond AS membership.

Table 4.6: Illustration of a Basic Address Profile for Addresses from AS 55430.

Address	Average Days Between Restarts	Last Restart Detected	Last Window	Days Since Last Restart
...				
A	29	05/04/14	12:00 - 17:59	88
B	28	05/04/14	12:00 - 17:59	88
C	13	05/04/14	12:00 - 17:59	88
D	16	07/17/14	00:00 - 05:59	14
E	12	07/17/14	00:00 - 05:59	14
F	8	06/22/14	00:00 - 05:59	39
...				

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Conclusions

This research sought to determine techniques to profile the behavior of network devices, specifically routers. The study focused on whether fragment identifiers could be used to remotely determine reboots and uptime of IPv6 addresses without remote access to the underlying systems. IPv6 routers were targeted because they are critical components of the IPv6 ecosystem. Understanding the behavior of these devices will provide insight into the behavior of IPv6 networks as a whole. More than 49,000 addresses were probed, four times daily, over a five month period. The TBT was used to cause addresses to send fragmented responses. The responses were then analyzed to extract the desired information.

The techniques presented in this study proved effective on 35% of the addresses probed. System restarts, operational time, and time between restarts could be inferred for these addresses. In addition, this research revealed a subset of addresses that appear to restart a large number of times relative to other addresses that indicated the presence of short lived infrastructure, possibly due to network optimization technologies.

This study revealed that approximately 87% of the 16,325 addresses analyzed either did not restart or restarted just once from March to July 2014. This observation suggests that the majority of known routers in IPv6 networks are stable, but rarely updated.

An exhaustive probe of all possible IPv6 addresses is infeasible, however, the techniques used in this study proved useful in creating behavioral profiles of known infrastructure through targeted probing.

5.1 Limitations

The techniques presented in this study rely on collecting sequential fragment identifiers. Approximately, 33% of the addresses tested returned random identifiers, 24% returned non-fragmented packets, and another 8% stopped responding to probing following the PTB message.

Although, operational time was determined for approximately 35% of the addresses tested,

this determination was limited to the time following the receipt of the first probe response. There was no way to calculate the length of time the addresses were active prior to the start of probing for this study. Due to the frequency of probing the operational time estimations had a possible variance of up to 6 hours. Additionally, the techniques used in this study would not detect multiple restarts between probing windows. Given that the fragment identifier for an address resets to the same value with each restart only the final restart within any window would be detected.

This research focused on detecting, and developing patterns of behavior based on restarts. The root cause of the restarts was not investigated. Differentiation between systems crashes and controlled restarts will be the subject of future work.

5.2 Future Work

This section presents a list for future research expanding on the techniques used in this study.

5.2.1 Use of Multiple Vantage Points

The use of multiple vantage points (i.e., probing systems in different physical locations) may help determine if the addresses from the No Fragments and No Response post PTB groups in Table 3.1 are being filtered at the host or along the path. Additionally, multiple vantage points will provide greater accuracy during the filtering phase.

Multiple vantage points will provide a redundancy in case of file corruption and system or path failure to a single vantage point.

5.2.2 BGP Correlation

The analysis of BGP updates may further verify the restart of a router. The correlation between the time at which a router restarts and a BGP update for the prefix containing that address can both validate that the router indeed restarted and identify the router as a peering router. Even though BGP updates may result from a number of events such as equipment failure, reconfiguration, or link failure [29], a BGP update that occurs at approximately the same time as an address' fragment identifier resetting would provide corroborating evidence of an actual device restart.

Additionally, multiple BGP updates observed between probing windows, in the absence of evidence of link failures, may help identify restarts missed by probing alone.

5.2.3 End-Host Analysis

Expanding Internet-wide testing to include end-host systems, Web and DSN servers for example, in future research will provide a more complete view of network device restart behavior. The frequency of security updates relative to other system updates will produce a clearer indication of the patching schedules for Internet accessible devices. These systems not only need to apply security updates related to their OS but also for any additional applications installed. Comparatively, these systems should restart more regularly than routers.

Also, end-host analysis may allow the restart algorithm to be validated against traffic that contain additional fragments. The addresses evaluated during this study only generated fragment identifiers as a response to the TBT; however, end-host systems may generate fragmented traffic as part of normal operation. This additional fragmented traffic will uncover possible discrepancies in the algorithm caused by multiple sources of fragmentation.

5.2.4 Root Cause Analysis

Analysis of the root cause of restarts will enhance the usefulness of the information revealed by this study. Understanding whether an anomalous restart is the result of a system crash, unscheduled maintenance, or malicious activity will provide security personnel and administrators valuable insight regarding the proper follow-on actions to pursue. Multiple system crashes may be an indication of imminent equipment failure, allowing administrators to preemptively replace the device.

Also, the ability to differentiate between maintenance-related restarts and system crashes would increase the accuracy of reliability and stability measurements. A device that restarts for scheduled maintenance would be categorized as more reliable than one the experiences a system crash.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Google. (2014, July) "IPv6 adoption"@ONLINE. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[2] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, "Measuring IPv6 adoption," ICSI Technical Report TR-13-004, Berkley, CA, Tech. Rep., August 2013.

[3] N. Sarrar, G. Maier, B. Ager, R. Sommer, and S. Uhlig, "Investigating IPv6 traffic - what happened at the world IPv6 day," in *Proceedings of the 13th International Conference on Passive and Active Network Measurement*, 2012.

[4] CAIDA. (2014, May) Archipelago measurement infrastructure@ONLINE. [Online]. Available: http://www.caida.org/projects/ark/

[5] APNIC. (2014, May) IPv6 measurements by economy organizational grouping and as number@ONLINE. [Online]. Available: http://labs.apnic.net/ipv6-measurement/

[6] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460 (Draft Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 5095, 5722, 5871, 6437, 6564, 6935, 6946, 7045, 7112. [Online]. Available: http://www.ietf.org/rfc/rfc2460.txt

[7] E. Durdağı and A. Buldu, "IPv4/IPv6 security and threat comparisons," *Procedia-Social and Behavioral Sciences*, vol. 2, no. 2, pp. 5285–5291, 2010.

[8] G. Blain, "Gov. Cuomo orders of probe Time Warner Cable internet outage," *NY Daily News*, August 2014. [Online]. Available: http://www.nydailynews.com/blogs/dailypolitics/gov-cuomo-orders-probe-time-warner-internet-outage-blog-entry-1.1918806

[9] Staff and Wire Reports, "Cuomo: NY to investigate Time Warner outage," *Democrat and Chronicle*, August 2014. [Online]. Available: http://www.democratandchronicle.com/story/news/2014/08/27/time-warner-cable-outages/14670903/

[10] R. Beverly, W. Brinkmeyer, M. Luckie, and J. P. Rohrer, "IPv6 alias resolution via induced fragmentation," in *Passive and Active Measurement*. Hong Kong: Springer Berlin, 2013, pp. 155–165.

[11] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the HTTPS certificate ecosystem," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. New York, NY, USA: ACM, 2013, pp. 291–304. [Online]. Available: http://doi.acm.org/10.1145/2504730.2504755

[12] V. Kundra, "Transition to IPv6," September 2010. [Online]. Available: http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/transition-to-ipv6.pdf

[13] (2011, February) The IANA IPv4 address free pool is now depleted ONLINE. [Online]. Available: https://www.arin.net/announcements/2011/20110203.html

[14] J. Brzozowski. (2012, April) IPv6 home networking pilot market deployment technical details. [Online]. Available: http://corporate.comcast.com/comcast-voices/ipv6-deployment-techonology-2

[15] L. Nobile. (2014, April) Arin enters phase four of the IPv4 countdown plan ONLINE. [Online]. Available: https://www.arin.net/announcements/2014/20140423.html

[16] J. Postel, "Internet Protocol," RFC 791 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1349, 2474, 6864. [Online]. Available: http://www.ietf.org/rfc/rfc791.txt

[17] J. McCann, S. Deering, and J. Mogul, "Path MTU Discovery for IP version 6," RFC 1981 (Draft Standard), Internet Engineering Task Force, Aug. 1996. [Online]. Available: http://www.ietf.org/rfc/rfc1981.txt

[18] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," RFC 2463 (Draft Standard), Internet Engineering Task Force, Dec. 1998, obsoleted by RFC 4443. [Online]. Available: http://www.ietf.org/rfc/rfc2463.txt

[19] A. Atlasis, "Attacking IPv6 implementation using fragmentation," *BlackHat Europe*, 2012.

[20] E. Nerakis, "IPv6 host fingerprint," DTIC Document, Ft. Belvoir, VA, Tech. Rep., 2006.

[21] R. Johari and D. K. H. Tan, "End-to-end congestion control for the internet: Delays and stability," *Networking, IEEE/ACM Transactions on*, vol. 9, no. 6, pp. 818–832, 2001.

[22] W. W.-K. Thong and G. Chen, "Jittering performance of random deflection routing in packet networks," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 3, pp. 616–624, 2013.

[23] V. E. Paxson, "Measurements and analysis of end-to-end internet dynamics," Ph.D. dissertation, University of California, Berkeley, 1997.

[24] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 133–145.

[25] K. Keys, Y. Hyun, M. Luckie, and K. Claffy, "Internet-scale IPv4 alias resolution with midar," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 2, pp. 383–399, 2013.

[26] A. Bender, R. Sherwood, and N. Spring, "Fixing ally's growing pains with velocity modeling," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, 2008, pp. 337–342.

[27] M. Luckie, R. Beverly, W. Brinkmeyer *et al.*, "Speedtrap: internet-scale IPv6 alias resolution," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 119–126.

[28] Z. Bischof and F. Bustamante, "A time for reliability - the growing importsnce of being always on," in *Proceedings of the ACM SIGCOMM'14*, 2014.

[29] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "Bgp routing stability of popular destinations," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. ACM, 2002, pp. 197–202.

[30] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," RFC 1323 (Proposed Standard), Internet Engineering Task Force, May 1992. [Online]. Available: http://www.ietf.org/rfc/rfc1323.txt

[31] (2014, June) Nmap OS detection@ONLINE. [Online]. Available: http://nmap.org/book/man-os-detection.html

[32] (2014, June) Netcraft. [Online]. Available: http://uptime.netcraft.com/accuracy.html#uptime

[33] (2014, May) What is GNS3?@ONLINE. [Online]. Available: http://www.gns3.net/

[34] M. Luckie, "Scamper: a scalable and extensible packet prober for active measurement of the internet," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 239–245.

[35] W. D. Brinkmeyer Jr, "IPv6 alias resolution via induced router fragmentation," DTIC Document, Ft. Belvoir, VA, Tech. Rep., 2013.

[36] (2014, March) The IPv6 topology dataset @ONLINE. [Online]. Available: http://www.caida.org/data/active/ipv6_allpref_topology_dataset.xml

[37] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291 (Draft Standard), Internet Engineering Task Force, Feb. 2006, updated by RFCs 5952, 6052, 7136. [Online]. Available: http://www.ietf.org/rfc/rfc4291.txt

[38] (2014, July) Team Cymru Community Services. [Online]. Available: http://www.team-cymru.org/Services/ip-to-asn.html

[39] (2014, June) GeoIP2 databases and services. [Online]. Available: https://www.maxmind.com/en/geolocation_landing

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia
2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California